SIAM J. OPTIM.
Vol. 23, No. 1, pp. 576–601

ON THE NONASYMPTOTIC CONVERGENCE OF CYCLIC COORDINATE DESCENT METHODS*

ANKAN SAHA† AND AMBUJ TEWARI‡

Abstract. Cyclic coordinate descent is a classic optimization method that has witnessed a resurgence of interest in signal processing, statistics, and machine learning. Reasons for this renewed interest include the simplicity, speed, and stability of the method, as well as its competitive performance on ℓ_1 regularized smooth optimization problems. Surprisingly, very little is known about its nonasymptotic convergence behavior on these problems. Most existing results either just prove convergence or provide asymptotic rates. We fill this gap in the literature by proving $O(1/k)$ convergence rates (where k is the iteration count) for two variants of cyclic coordinate descent under an isotonicity assumption. Our analysis proceeds by comparing the objective values attained by the two variants with each other, as well as with the gradient descent algorithm. We show that the iterates generated by the cyclic coordinate descent methods remain better than those of gradient descent uniformly over time.

Key words. convex optimization, cyclic coordinate descent, convergence rates, sparsity

AMS subject classifications. 90C06, 90C25

DOI. 10.1137/110840054

1. Introduction.

As we encounter larger and higher dimensional datasets, we are faced with novel challenges in designing and analyzing optimization algorithms that can work efficiently with such datasets. This paper considers one such class of algorithms, namely, *cyclic coordinate descent* and variants thereof. There has been recent work demonstrating the potential of these algorithms for solving large and high-dimensional ℓ_1 regularized loss minimization problems:

$$\min_x \ \frac{1}{n}\sum_{i=1}^{n} \ell(x, Z_i) + \lambda\|x\|_1 \ , \tag{1.1}$$

where $x \in \mathbb{R}^d$ is a possibly high-dimensional predictor that is being estimated from the samples $Z_i = (X_i, Y_i)$ consisting of input and output pairs, ℓ is a convex loss function measuring prediction performance, and $\lambda \geq 0$ is a "regularization" parameter. The use of the ℓ_1 norm $\|x\|_1$ (sum of absolute values of x_i) as a "penalty" or "regularization term" is motivated by its sparsity-promoting properties, and there is a large and growing literature studying such issues (see, e.g., [5, 9] and the references therein). In this paper, we restrict ourselves to analyzing the behavior of coordinate descent methods on problems like (1.1) above. The general idea behind coordinate descent is to choose, at each iteration, an index j and change x_j such that the objective function decreases. Choosing j can be as simple as cycling through the coordinates, or a more sophisticated coordinate selection rule can be employed. Hastie and coauthors [10, 11] use the cyclic rule, which is also the one we analyze in this paper. From their empirical comparisons on simulated and real datasets, they conclude the following regarding

*Received by the editors July 8, 2011; accepted for publication (in revised form) November 14, 2012; published electronically March 28, 2013.
http://www.siam.org/journals/siopt/23-1/84005.html
†Department of Computer Science, University of Chicago, Chicago, IL 60637 (ankans@cs.uchicago.edu).
‡Department of Statistics, University of Michigan, Ann Arbor, MI 48109 (tewaria@umich.edu).

cyclic coordinate descent [11]: "*Its computational speed both for large N and p are quite remarkable.*"[1]

It is natural to attempt an analysis that can provide theoretical support for the good performance of cyclic coordinate descent on the ℓ_1 regularized loss minimization problem (1.1). Our goal is to do this by obtaining *nonasymptotic* rates of convergence, i.e., guarantees about accuracy of iterative optimization algorithms that hold right from the first iteration. This is in contrast to asymptotic guarantees that only hold once the iteration count is "large enough" (and often what is meant by "large enough" is not explicitly quantified). For our analysis, we abstract away the particulars of the setting above and view (1.1) as a special case of the convex optimization problem:

$$\min_{x \in \mathbb{R}^d} \; F(x) := f(x) + \lambda \|x\|_1 \; . \tag{1.2}$$

In order to obtain nonasymptotic convergence rates, one must assume that f is "nice" in some sense. This can be formalized in different ways: for instance, by making assumptions of Lipschitz continuity, differentiability, or strong convexity. We will assume that f is differentiable with a Lipschitz continuous gradient. In the context of problem (1.1), it suffices to assume that the loss ℓ is twice differentiable with a bounded second derivative. Several losses, such as squared loss and logistic loss, satisfy this condition. Our results therefore apply to ℓ_1 regularized least squares[2] and to ℓ_1 regularized logistic regression.

For a method as old as cyclic coordinate descent, it is surprising that little is known about nonasymptotic convergence even under smoothness assumptions. As far as we know, nonasymptotic results are not available even when $\lambda = 0$, i.e., for the unconstrained smooth convex minimization problem. Given recent empirical successes of the method, we feel that this gap in the literature needs to be filled urgently. In fact, this sentiment is shared by Wu and Lange [25], who lamented, "*Better understanding of the convergence properties of the algorithms is sorely needed.*" They were talking about greedy coordinate descent methods, but their comment applies to cyclic methods as well.

The situation with gradient descent methods is much better. There are a variety of nonasymptotic convergence results available in the literature (see, for example, [16]). Our strategy in this paper is to leverage these results to shed some light on the convergence of coordinate descent methods. We do this via a series of *comparison theorems* that relate variants of coordinate descent methods to each other and to the gradient descent algorithm. To do this, we make assumptions on the starting point and an additional *isotonicity* assumption on the gradient of the function f. Since nonasymptotic $O(1/k)$ accuracy guarantees are available for gradient descent, we are able to prove the same rates for two variants of cyclic coordinate descent. Here k is the iteration count, and the constants hidden in the $O(\cdot)$ notation are small and known. Nesterov [17] has remarked that it is "*almost impossible to estimate the rate of convergence*" of cyclic coordinate descent "*in the general case.*" We therefore feel that establishing rates under particular assumptions is a small but important step towards a full understanding of these methods. Of course, the next step is to relax, or even eliminate, the additional assumptions we make (these are detailed in section 4), and doing this is an important open problem left for future work. Our experimental

[1] Their N and p refer to number of samples and dimensions respectively, i.e., n and d in our notation.

[2] This is known as "lasso" in machine learning and statistics and "basis pursuit" in signal processing.

results in section 9 make us confident that nonasymptotic guarantees will eventually be proven for cyclic coordinate descent under fewer or no restrictive assumptions.

We find it important to state at the outset that our aim here is not to give the best possible rates for the problem (1.2). For example, even among gradient-based methods, faster $O(1/k^2)$ nonasymptotic accuracy bounds can be achieved using Nesterov's celebrated method [15] or its later variants. Instead, our goal is to better understand cyclic coordinate descent methods and their relationship to gradient descent.

1.1. Related work. Coordinate descent methods are quite old, and we cannot attempt a survey here. Instead, we refer the reader to the works of Tseng and coauthors [22, 23, 24] that summarize previous work and also present analyses for coordinate descent methods. These consider cyclic coordinate descent as well as versions that use more sophisticated coordinate selection rules. However, as mentioned above, the analyses either establish convergence without rates or give asymptotic rates that hold after sufficiently many iterations have occurred. An exception is [23], which does give nonasymptotic rates but for a version of coordinate descent that is not cyclic. However, the topic seems to have recently caught the attention of several researchers, and there are now a handful of other papers [4, 7, 17, 19, 20, 21] that provide nonasymptotic convergence rates for coordinate descent algorithms that can be applied to the problem (1.1). We will discuss these papers in section 10.

We mentioned that the empirical success reported in [10] was our motivation to consider cyclic coordinate descent for ℓ_1 regularized problems. They consider the lasso problem:

$$\min_{x \in \mathbb{R}^d} \ \frac{1}{2}\|Xx - y\|^2 + \lambda \|x\|_1 \ , \tag{1.3}$$

where $X \in \mathbb{R}^{n \times d}$ and $y \in \mathbb{R}^n$. In this case, the smooth part f is a quadratic,

$$f(x) = \tfrac{1}{2} \langle Ax, x \rangle + \langle b, x \rangle \ , \tag{1.4}$$

where $A = X^\top X$ and $b = -X^\top y$. Note that A is symmetric and positive semidefinite. Cyclic coordinate descent has also been applied to the ℓ_1 regularized logistic regression problem [12]. Since the logistic loss is twice differentiable, this problem also falls within the class of problems considered in this paper.

1.2. Outline. Notation and necessary definitions are given in section 2. The gradient descent algorithm and two variants of cyclic coordinate descent are presented in section 3. Section 4 spells out the additional assumptions on f that our current analysis needs. Section 5 proves a monotonicity result about gradient descent iterates. Sections 6 and 7 do the same for iterates generated by the two variants of coordinate descent. Additionally, these two sections prove results comparing the iterates generated by the three algorithms when they are all started from the same point. Similar comparison theorems in the context of solving a system of nonlinear equations using Jacobi and Gauss–Seidel methods appear in a paper of Rheinboldt [18]. The comparison theorems set the stage for the main results given in section 8.

Section 8 converts the comparison between iterates into a comparison between objective function values achieved by the iterates. The nonasymptotic convergence rates of cyclic coordinate descent are then inferred from rates for gradient descent. The main question that remains open after our analysis is whether cyclic coordinate descent can be shown to enjoy nonasymptotic convergence guarantees with fewer or perhaps no extra assumptions. Experiments with simulated data in section 9 suggest

that the answer should be affirmative. Section 10 further discusses avenues for future exploration and situates our work in the context of existing work on nonasymptotic guarantees for coordinate descent methods.

2. Preliminaries and notation. We use the lowercase letters x, y, z, g, and γ to refer to vectors throughout the paper. Letters with parenthesized superscripts, like $x^{(k)}$, refer to vectors as well, whereas we use subscripts, as in x_j, to refer to the components of vectors. A numerical constant in bold, such as $\mathbf{1}$, refers to a vector all of whose entries are 1.

For any positive integer k, $[k] := \{1, \ldots, k\}$. By $\operatorname{sign}(a)$, we mean the *interval-valued* sign function defined as

$$\operatorname{sign}(a) := \begin{cases} \{-1\} & \text{if } a < 0, \\ \{1\} & \text{if } a > 0, \\ [-1, 1] & \text{if } a = 0. \end{cases}$$

Unless otherwise specified, $\|\cdot\|$ refers to the Euclidean norm $\|x\| := \left(\sum_i x_i^2\right)^{\frac{1}{2}}$, $\|\cdot\|_1$ denotes the l_1 norm $\|x\|_1 := (\sum_i |x_i|)$, and $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product $\langle x, y \rangle := \sum_i x_i y_i$. Throughout the paper, inequalities between vectors are to be interpreted componentwise, i.e., for $x, y \in \mathbb{R}^d$, $x \geq y$ means that $x_i \geq y_i$ for all $i \in [d]$. The following definition will be used extensively in the paper.

DEFINITION 2.1. *Suppose a function $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable on $\mathbb{R}^d$. Then f is said to have Lipschitz continuous gradient with respect to the Euclidean norm $\|\cdot\|$ if there exists a constant L such that*

$$\|\nabla f(x) - \nabla f(x')\| \leq L\|x - x'\| \qquad \forall\ x, x' \in \mathbb{R}^d. \tag{2.1}$$

An important fact (see, e.g., [16, Thm. 2.1.5]) we will use is that if a function f has Lipschitz continuous gradient with respect to a norm $\|\cdot\|$, then it satisfies the following generalized bounded Hessian property:

$$\forall x, x' \in \mathbb{R}^d, \qquad f(x) \leq f(x') + \langle \nabla f(x'), x - x' \rangle + \frac{L}{2}\|x - x'\|^2. \tag{2.2}$$

An operator $T : \mathbb{R}^d \to \mathbb{R}$ is said to be *isotone* iff

$$x \geq y \quad \Rightarrow \quad T(x) \geq T(y). \tag{2.3}$$

An important isotone operator that we will frequently deal with is the *shrinkage* operator $\mathbf{S}_\tau : \mathbb{R}^d \to \mathbb{R}$ defined, for $\tau > 0$, as

$$[\mathbf{S}_\tau(x)]_i := S_\tau(x_i), \tag{2.4}$$

where $S_\tau(a)$ is the scalar shrinkage operator:

$$S_\tau(a) := \begin{cases} a - \tau, & a > \tau, \\ 0, & a \in [-\tau, \tau], \\ a + \tau, & a < -\tau. \end{cases} \tag{2.5}$$

Algorithm 1: Gradient descent (GD).

Initialize: Choose an appropriate initial point $x^{(0)}$.
for $k = 0, 1, \ldots$ **do**
  $x^{(k+1)} \leftarrow \mathbf{S}_{\lambda/L}(x^{(k)} - \frac{\nabla f(x^{(k)})}{L})$
end for

3. Algorithms. We will consider three iterative algorithms for solving the minimization problem (1.2). All of them enjoy the descent property $F(x^{(k+1)}) \leq F(x^{(k)})$ for successive iterates $x^{(k)}$ and $x^{(k+1)}$.

Algorithm 1, a simple extension of gradient descent (GD), is one of the most common iterative algorithms used for convex optimization (see [1, 8] and the references therein). It is based on the idea that, using (2.2), we can come up with the following global upper approximation of F:

$$F(x) \leq f(x^{(k)}) + \left\langle \nabla f(x^{(k)}), x - x^{(k)} \right\rangle + \frac{L}{2}\|x - x^{(k)}\|^2 + \lambda\|x\|_1 \ .$$

This approximation is exact at $x = x^{(k)}$. It is easy to show [1] that the above approximation is minimized at $x = \mathbf{S}_{\lambda/L}(x^{(k)} - \nabla f(x^{(k)})/L)$. This is the next iterate for the GD algorithm. We call it "gradient descent" as it reduces to the algorithm

$$x^{(k+1)} = x^{(k)} - \frac{\nabla f(x^{(k)})}{L}$$

when there is no regularization (i.e., $\lambda = 0$). Finite time convergence rates for the GD algorithm are well known.

THEOREM 3.1. *Let* $\left\{x^{(k)}\right\}$ *be the sequence generated by the* GD *algorithm. Then, for any minimizer* $x^\star$ *of* (1.2)*, and for all* $k \geq 1$*,*

$$F(x^{(k)}) - F(x^\star) \leq \frac{L\|x^\star - x^{(0)}\|^2}{2k} \ .$$

The above theorem can be found in, e.g., [1, Theorem 3.1].

The second algorithm, cyclic coordinate descent (CCD), instead of using the current gradient to update all components simultaneously, goes through them in a cyclic fashion. The next "outer" iterate $y^{(k+1)}$ is obtained from $y^{(k)}$ by creating a

Algorithm 2: Cyclic coordinate descent (CCD).

Initialize: Choose an appropriate initial point $y^{(0)}$.
for $k = 0, 1, \ldots$ **do**
  $y^{(k,0)} \leftarrow y^{(k)}$
  for $j = 1$ to d **do**
    $y_j^{(k,j)} \leftarrow S_{\lambda/L}(y_j^{(k,j-1)} - [\nabla f(y^{(k,j-1)})]_j \,/\, L)$
    $\forall i \neq j,\ y_i^{(k,j)} \leftarrow y_i^{(k,j-1)}$
  end for
  $y^{(k+1)} \leftarrow y^{(k,d)}$
end for

series of d intermediate or "inner" iterates $y^{(k,j)}$, $j \in [d]$, where $y^{(k,j)}$ differs from $y^{(k,j-1)}$ only in the jth coordinate, whose value can be found by minimizing the following one-dimensional overapproximation of F over the scalar α:

$$f(y^{(k,j-1)}) + \lambda \sum_{i \neq j} |y_i^{(k,j-1)}| + [\nabla f(y^{(k,j-1)})]_j \cdot (\alpha - y_j^{(k,j-1)}) + \frac{L}{2}(\alpha - y^{(k,j-1)})_j^2 + \lambda|\alpha| \ . \tag{3.1}$$

It can again be verified that the above minimization has the closed form solution

$$\alpha = S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) \ ,$$

which is what CCD chooses $y_j^{(k,j)}$ to be. Once all coordinates have been cycled through, $y^{(k+1)}$ is simply set to be $y^{(k,d)}$. Let us point out that in an actual implementation, the inner iterates $y^{(k,j)}$ would not be computed separately, but $y^{(k)}$ would be updated "in place." For analysis purposes, it is convenient to give names to the intermediate iterates. Note that for all $j \in \{0, 1, \ldots, d\}$ the inner iterate looks like

$$y^{(k,j)} = \left[y_1^{(k+1)}, \ldots, y_j^{(k+1)}, y_{j+1}^{(k)}, \ldots, y_d^{(k)}\right] \ .$$

When updating the jth coordinate, CCD uses the newer gradient value $\nabla f(y^{(k,j-1)})$ rather than $\nabla f(y^{(k)})$, which is used by GD. We might hope that CCD converges faster than GD due to the use of "fresh" information. Therefore, it is natural to expect that CCD should enjoy the nonasymptotic convergence rate given in Theorem 3.1 (or better). We show this is indeed the case under an *isotonicity assumption* stated in section 4 below. Under the assumption, we are actually able to show the correctness of the intuition that CCD should converge faster than GD.

The third and final algorithm that we consider is cyclic coordinate minimization (CCM). The only way it differs from CCD is that instead of minimizing the one-dimensional overapproximation (3.1), it chooses $z_j^{(k,j)}$ to minimize

$$F(z_1^{(k,j-1)}, \ldots, z_{j-1}^{(k,j-1)}, \alpha, z_{j+1}^{(k,j-1)}, \ldots, z_d^{(k,j-1)})$$

over α. In a sense, CCM is not actually an algorithm as it does not specify how to minimize F for any arbitrary smooth function f. An important case when the

Algorithm 3: Cyclic coordinate minimization (CCM).

Initialize: Choose an appropriate initial point $z^{(0)}$.
for $k = 0, 1, \ldots$ **do**
  $z^{(k,0)} \leftarrow z^{(k)}$
  for $j = 1$ to d **do**
    $z_j^{(k,j)} \leftarrow \operatorname{argmin}_\alpha \ F(z_1^{(k,j-1)}, \ldots, z_{j-1}^{(k,j-1)}, \alpha, z_{j+1}^{(k,j-1)}, \ldots, z_d^{(k,j-1)})$
    $\forall i \neq j$, $z_i^{(k,j)} \leftarrow z_i^{(k,j-1)}$
  end for
  $z^{(k+1)} \leftarrow z^{(k,d)}$
end for

minimum can be computed exactly is when f is quadratic as in (1.4). In that case, we have

$$z_j^{(k,j)} = S_{\lambda/A_{j,j}}\left(z_j^{(k,j-1)} - \frac{[Az^{(k,j-1)} + b]_j}{A_{j,j}}\right).$$

If there is no closed form solution, then we might have to resort to numerical minimization in order to implement CCM. This is usually not a problem since one-dimensional convex functions can be minimized numerically to an extremely high degree of accuracy in a few steps (of, say, the Newton method). For the purpose of analysis, we will assume that an exact minimum is found. Again, intuition suggests that the accuracy of CCM after any fixed number of iterations should be better than that of CCD since CCD only minimizes an overapproximation. Under the same isotonicity assumption that we mentioned above, we can show that this intuition is indeed correct.

We end this section with a cautionary remark regarding terminology. In the literature, CCM appears much more frequently than CCD, and it is actually the former that is often referred to as "cyclic coordinate descent" (see [10] and the references therein). Our reasons for considering CCD are (i) it is a nice, efficient alternative to CCM, and (ii) a stochastic version of CCD (where the coordinate to update is chosen randomly and not cyclically) is already known to enjoy a nonasymptotic $O(1/k)$ expected convergence rate [17, 19, 21].

4. Isotonicity, supersolutions, and subsolutions. We already mentioned the known convergence rate for GD (Theorem 3.1) above. Before delving into the analysis, it is necessary to state an assumption on f that, once appropriate starting conditions are imposed, results in particularly interesting properties of the convergence behavior of GD, as described in Proposition 5.1. The GD algorithm generates iterates by applying the operator

$$T_{GD}(x) := \mathbf{S}_{\lambda/L}\left(x - \frac{\nabla f(x)}{L}\right) \tag{4.1}$$

repeatedly. It turns out that if T_{GD} is an isotone operator, then the GD iterates satisfy the properties claimed in Proposition 5.1, which is essential for our convergence analysis. The above operator is a composition of $\mathbf{S}_{\lambda/L}$, an isotone operator, and $\mathbf{I} - \nabla f/L$ (where $\mathbf{I}$ denotes the identity operator). To ensure overall isotonicity, it suffices to assume that $\mathbf{I} - \nabla f/L$ is isotone. We formally state this as an assumption.

Assumption 1. The operator $x \mapsto x - \frac{\nabla f(x)}{L}$ is isotone.

Similar assumptions appear in the literature comparing Jacobi and Gauss–Seidel methods for solving linear equations. For example, the *Stein–Rosenberg theorem* [3, Chapter 2] holds under these assumptions. When the function f is quadratic as in (1.4), our assumption is equivalent to assuming that the off-diagonal entries in A are nonpositive, i.e., $A_{i,j} \leq 0$ for all $i \neq j$. For a general smooth f, the following condition is sufficient to make the assumption true: f is twice differentiable and the Hessian $\nabla^2 f(x)$ at any point x has nonpositive off-diagonal entries.

There are many examples of such matrices A which are of interest. In particular, graph Laplacians given by $L = D - W$ (where W refers to the weighted adjacency graph and D is a diagonal matrix given by $D_{i,i} = \sum_j W_{i,j}$) have nonpositive off-diagonal entries. Minimization of the corresponding quadratic objective forms the basis of spectral partitioning [6] as well as semisupervised learning [2]. We note that matrices with positive diagonal and nonpositive off-diagonal entries occur frequently

enough in matrix analysis to have a name: they are called *Minkowski–Metzler* matrices [14].

In the next few sections, we will see how the isotonicity assumption leads to an isotonically decreasing (or increasing) behavior of GD, CCD, and CCM iterates under appropriate starting conditions. To specify what these starting conditions are, we need the notions of super- and subsolutions.

DEFINITION 4.1. *A vector x is a supersolution iff, for some $\tau > 0$,*

$$x \geq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right) .$$

Analogously, x is a subsolution iff, for some $\tau > 0$,

$$x \leq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right) .$$

Since the inequalities above are vector inequalities, an arbitrary x may be neither a supersolution nor a subsolution. The names "supersolution" and "subsolution" are justified because equality holds in the definitions above $\left(\text{i.e., } x = \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right)\right)$ iff x is a minimizer of F. To see this, note that subgradient optimality conditions say that x is a minimizer of $F = f + \lambda\|\cdot\|_1$ iff for all $j \in [d]$

$$0 \in [\nabla f(x)]_j + \lambda \operatorname{sign}(x_j) . \tag{4.2}$$

Further, it is easy to see that

$$\forall a, b \in \mathbb{R},\ \tau > 0, 0 \in b + \lambda \operatorname{sign}(a) \ \Leftrightarrow\ a = S_{\lambda/\tau}(a - b/\tau) . \tag{4.3}$$

From the definition of super- and subsolutions, it is obvious that a solution (in other words, a minimizer of F) is both a super- and subsolution. If that was all, then any result that holds under the assumption that the starting point is a super- or subsolution would be vacuous. To show that the super- and subsolution concepts are not degenerate, we provide the following result.

LEMMA 4.2. *A sufficient condition for x to be a supersolution (resp., subsolution) is $\nabla f(x) \geq \lambda\mathbf{1}$ (resp., $\nabla f(x) \leq -\lambda\mathbf{1}$).*

Proof. Since $y + \lambda\mathbf{1} \geq \mathbf{S}_\lambda(y)$ for any y, we have

$$\nabla f(x) \geq \lambda\mathbf{1} \ \Rightarrow\ x \geq x - \nabla f(x) + \lambda\mathbf{1} \ \Rightarrow\ x \geq \mathbf{S}_\lambda(x - \nabla f(x)) .$$

The proof for the subsolution case is similar. ∎

Consider the quadratic case again. That is, the function f is as in (1.4). The above lemma implies that if x satisfies $Ax + b > \lambda\mathbf{1}$, then it is a supersolution. If we can find an x such that $Ax > \mathbf{0}$, then, after a possible scaling, we can satisfy $Ax > \lambda\mathbf{1} - b$ and hence find a supersolution. Moreover, this discussion also shows that supersolutions are available in abundance if A is nonsingular since $Ax = y$ has a solution for every $y > \mathbf{0}$. We mentioned Minkowski–Metzler matrices above. If a Minkowski–Metzler matrix is also diagonally dominant, then we have $A\mathbf{1} > 0$, and hence $\mathbf{1}$ can be scaled to produce a supersolution. Recall that a matrix A is said to be (strictly row) diagonally dominant whenever

$$\forall i,\ |A_{i,i}| > \sum_{j \neq i} |A_{i,j}| .$$

It is interesting to note that, in the context of solving a system of linear equations, diagonally dominant matrices play a crucial role in ensuring the convergence of iterative methods [3, Chapter 2].

Now we prove a couple of properties of super- and subsolutions that will prove useful later. The first is a scale invariance property of super- and subsolutions. This scale invariance with respect to τ means that we can replace the quantifier "for some $\tau > 0$" to "for all $\tau > 0$" in Definition 4.1 without changing the concept being defined.

LEMMA 4.3. *If the inequality*

$$x \geq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right) \tag{4.4}$$

holds for some $\tau > 0$, then it holds for all $\tau > 0$.

Similarly, if the inequality

$$x \leq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right)$$

holds for some $\tau > 0$, then it holds for all $\tau > 0$.

Proof. We provide the proof for the supersolution case only, as the proof in the subsolution case is analogous. Suppose, for a particular $\tau > 0$, we have

$$x \geq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right) .$$

Fix an arbitrary coordinate j and recall that S is the scalar shrinkage operator. The remainder of the proof is divided into arguments dealing with three disjoint cases depending upon the value of $[\nabla f(x)]_j$.

Case 1: $[\nabla f(x)]_j - \lambda > 0$. This case is illustrated in Figure 4.1. As $\tau > 0$ changes, the graph of the scalar function

$$p \mapsto S_{\lambda/\tau}\left(p - \frac{[\nabla f(x)]_j}{\tau}\right) \tag{4.5}$$

changes, but it is clear that division by τ does not alter the relative ordering of the values attained by the above function and the identity function. As is evident from Figure 4.1, the graph of $p \mapsto p$ always lies above that of the function (4.5) above. Thus

$$x_j \geq S_{\lambda/\tau}\left(x_j - \frac{[\nabla f(x)]_j}{\tau}\right) \tag{4.6}$$

for all values of $\tau > 0$.

Case 2: $0 \in [[\nabla f(x)]_j - \lambda, [\nabla f(x)]_j + \lambda]$. This case is illustrated in Figure 4.2. It is clear from the figure that, for $\tau > 0$, we have $x_j \geq S_{\lambda/\tau}\big(x_j - \frac{[\nabla f(x)]_j}{\tau}\big)$ iff $x_j \geq 0$. Just as in the previous case, changing the value of τ does not alter the relative ordering of the values attained by the function (4.5) and the identity function. Thus (4.6) holds iff $x_j \geq 0$ irrespective of the value of $\tau > 0$.

Case 3: $[\nabla f(x)]_j + \lambda < 0$. As illustrated in Figure 4.3, in this case the graph of the function (4.5) always lies below that of the identity function. Thus (4.6) will not be satisfied for any value of τ, which makes this case vacuous.

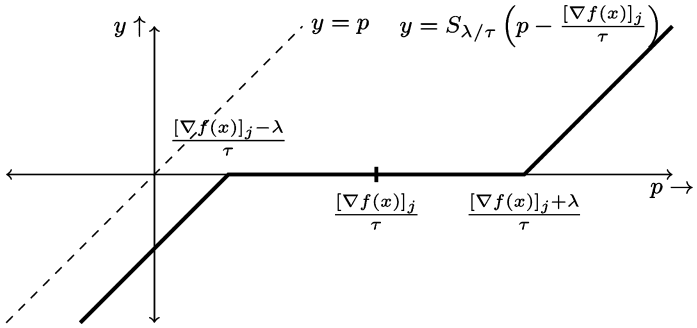


FIG. 4.1. *Interval to the right of zero.*

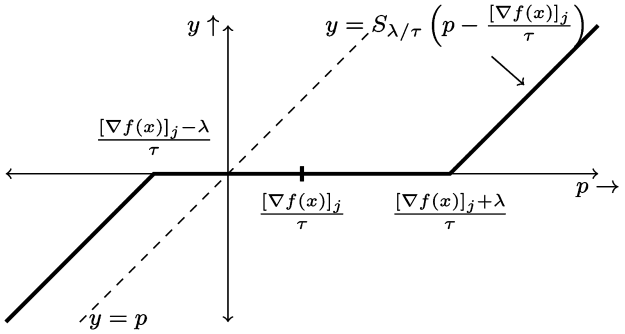


FIG. 4.2. *Interval crossing zero.*

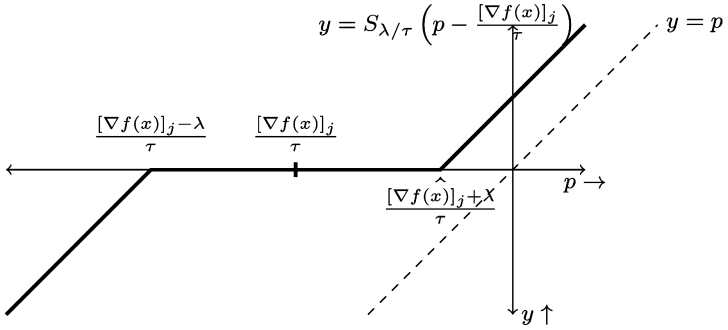


FIG. 4.3. *Interval to the left of zero.*

Therefore, in all three cases, whether or not the inequality (4.6) holds is independent of the value of $\tau > 0$. □

The second property is the monotonicity of a certain function of a single variable.

LEMMA 4.4. *If x is a supersolution (resp., subsolution), then for any j, the function*

$$\tau \mapsto S_{\lambda/\tau}\left(x_j - \frac{[\nabla f(x)]_j}{\tau}\right)$$

is monotonically nondecreasing (resp., nonincreasing) on $(0, \infty)$.

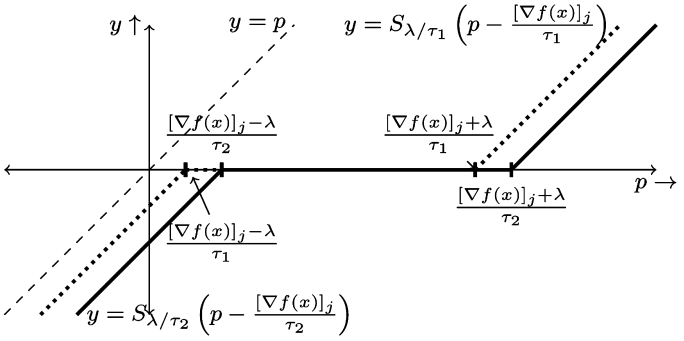


FIG. 4.4. *Interval to the right of zero.*

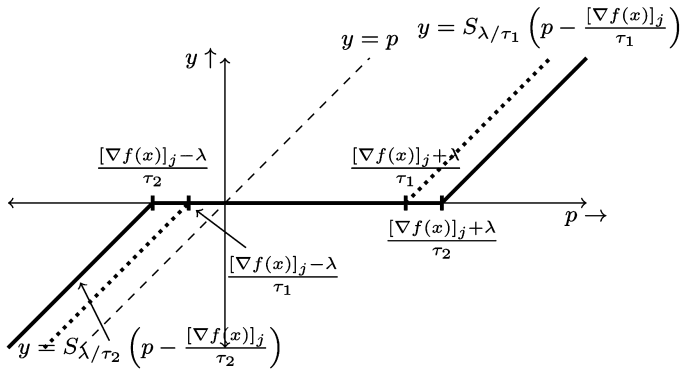


FIG. 4.5. *Interval crossing zero.*

Proof. Let

$$h(\tau) = S_{\lambda/\tau}\left(x_j - \frac{[\nabla f(x)]_j}{\tau}\right) .$$

We again look at the three disjoint cases for arbitrary $\tau_1, \tau_2 \in (0, \infty)$ with $\tau_1 \geq \tau_2$.

Case 1: $[\nabla f(x)]_j - \lambda > 0$. Note that both the hinge points in the graph of the functions

$$p \mapsto S_{\lambda/\tau_i}\left(p - \frac{[\nabla f(x)]_j}{\tau_i}\right),\ i \in \{1, 2\},$$

will be positive (Figure 4.4). Further, we have

$$\frac{[\nabla f(x)]_j - \lambda}{\tau_1} \leq \frac{[\nabla f(x)]_j - \lambda}{\tau_2}\ , \qquad \frac{[\nabla f(x)]_j + \lambda}{\tau_1} \leq \frac{[\nabla f(x)]_j + \lambda}{\tau_2}\ .$$

Thus, it is easy to see that $h(\tau_1)$ is greater than $h(\tau_2)$.

Case 2: $0 \in [[\nabla f(x)]_j - \lambda, [\nabla f(x)]_j + \lambda]$. Since x needs to be a supersolution, we only need to consider the subset of the domain when $x_j \geq 0$. We still have $\frac{[\nabla f(x)]_j + \lambda}{\tau_1} \leq \frac{[\nabla f(x)]_j + \lambda}{\tau_2}$, and it is obvious from Figure 4.5 that $h(\tau_1) \geq h(\tau_2)$.

Case 3: $[\nabla f(x)]_j + \lambda < 0$. Since x can never be a supersolution in this case as shown in the proof of Lemma 4.3, this case is vacuous.

Thus, we have shown that $h(\tau_1) \geq h(\tau_2)$ in all three cases. ∎

5. Gradient descent. The next result says that, under Assumption 1, GD generates iterates whose components decrease (resp., increase) monotonically when started from a supersolution (resp., subsolution).

PROPOSITION 5.1. *Suppose Assumption* 1 *holds. If* $x^{(0)}$ *is a supersolution and* $\{x^{(k)}\}$ *is the sequence of iterates generated by the* GD *algorithm, then for all* $k \geq 0$

$$(1) \quad x^{(k+1)} \leq x^{(k)}, \qquad (2) \quad x^{(k)} \text{ is a supersolution.}$$

If $x^{(0)}$ *is a subsolution and* $\{x^{(k)}\}$ *is the sequence of iterates generated by the* GD *algorithm, then for all* $k \geq 0$

$$(1) \quad x^{(k+1)} \geq x^{(k)}, \qquad (2) \quad x^{(k)} \text{ is a subsolution.}$$

Proof. We only prove the supersolution case. The proof for the subsolution case is analogous. Consider the operator

$$T_{GD}(x) := \mathbf{S}_{\lambda/L}\left(x - \frac{\nabla f(x)}{L}\right)$$

given by (4.1). By the isotonicity assumption, T_{GD} is an isotone operator. We will prove by induction that $T_{GD}(x^{(k)}) \leq x^{(k)}$. This proves that $x^{(k+1)} \leq x^{(k)}$ since $x^{(k+1)} = T_{GD}(x^{(k)})$. Using Lemma 4.3, the second claim follows by the definition of the T_{GD} operator.

The base case $T_{GD}(x^{(0)}) \leq x^{(0)}$ is true by Lemma 4.3 since $x^{(0)}$ is given to be a supersolution. Now assume $T_{GD}(x^{(k)}) \leq x^{(k)}$. Applying the isotone operator T_{GD} on both sides, we get $T_{GD}(T_{GD}(x^{(k)})) \leq T_{GD}(x^{(k)})$. This is the same as $T_{GD}(x^{(k+1)}) \leq x^{(k+1)}$ by the definition of $x^{(k+1)}$, which completes our inductive claim. ∎

6. Cyclic coordinate descent. Now, we prove a result for CCD that is analogous to Proposition 5.1. However, this time a little more work is involved.

PROPOSITION 6.1. *Suppose Assumption* 1 *holds. If* $y^{(0)}$ *is a supersolution and* $\{y^{(k)}\}$ *is the sequence of iterates generated by the* CCD *algorithm, then for all* $k \geq 0$

$$(1) \quad y^{(k+1)} \leq y^{(k)}, \qquad (2) \quad y^{(k)} \text{ is a supersolution.}$$

If y_0 *is a subsolution and* $\{y^{(k)}\}$ *is the sequence of iterates generated by the* CCD *algorithm, then for all* $k \geq 0$

$$(1) \quad y^{(k+1)} \geq y^{(k)}, \qquad (2) \quad y^{(k)} \text{ is a subsolution.}$$

Proof. We will only give the proof for the supersolution case, as the proof for the subsolution case is similar. We start with a supersolution $y^{(0)}$. We will prove the following: if $y^{(k)}$ is a supersolution, then

$$y^{(k+1)} \leq y^{(k)} \,, \tag{6.1}$$

$$y^{(k+1)} \text{ is a supersolution.} \tag{6.2}$$

Then the lemma follows by induction on k. Let us make the induction assumption that $y^{(k)}$ is a supersolution and try to prove (6.1) and (6.2). To prove these, we will show that $y^{(k,j)} \leq y^{(k)}$ and $y^{(k,j)}$ is a supersolution by (a second or inner) induction on $j \in \{0, 1, \ldots, d\}$. This proves (6.1) and (6.2) for $y^{(k+1)}$ since $y^{(k+1)} = y^{(k,d)}$.

For the base case ($j = 0$) of the (inner) induction, note that $y^{(k,0)} \leq y^{(k)}$ is trivial since the two vectors are equal. For the same reason, $y^{(k,0)}$ is a supersolution since we have assumed $y^{(k)}$ to be a supersolution. Now assume $y^{(k,j-1)} \leq y^{(k)}$ and $y^{(k,j-1)}$ is a supersolution for some $j > 0$. We want to show that $y^{(k,j)} \leq y^{(k)}$ and $y^{(k,j)}$ is a supersolution.

Since $y^{(k,j-1)}$ and $y^{(k,j)}$ differ only in the jth coordinate, to show that $y^{(k,j)} \leq y^{(k)}$ given $y^{(k,j-1)} \leq y^{(k)}$, it suffices to show that $y^{(k,j)} \leq y^{(k,j-1)}$, i.e.,

$$y_j^{(k,j)} \leq y_j^{(k,j-1)} = y_j^{(k)} \ . \tag{6.3}$$

Since $y^{(k,j-1)} \leq y^{(k)}$, applying the isotone operator $\mathbf{I} - \nabla f/L$ on both sides and taking the jth coordinate gives

$$y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L} \leq y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L} \ .$$

Applying the scalar shrinkage operator on both sides gives

$$S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) \leq S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L}\right) \leq y_j^{(k)} \ .$$

The left-hand side is $y_j^{(k,j)}$ by definition, while the second inequality follows because $y^{(k)}$ is a supersolution. Thus, we have proved (6.3).

Now we prove that $y^{(k,j)}$ is a supersolution. Note that we have already shown $y^{(k,j)} \leq y^{(k,j-1)}$. Applying the isotone operator $\mathbf{I} - \frac{\nabla f}{L}$ on both sides gives

$$y_j^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_j}{L} \leq y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L} \ , \tag{6.4}$$

$$\forall i \neq j,\ y_i^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_i}{L} \leq y_i^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_i}{L} \ . \tag{6.5}$$

Applying a scalar shrinkage on both sides of (6.4) and noting that the right-hand side is $y_j^{(k,j)}$ by definition, we have

$$S_{\lambda/L}\left(y_j^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_j}{L}\right) \leq y_j^{(k,j)} \ . \tag{6.6}$$

For $i \neq j$, we have

$$\begin{aligned} y_i^{(k,j)} = y_i^{(k,j-1)} &\geq S_{\lambda/L}\left(y_i^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_i}{L}\right) \\ &\geq S_{\lambda/L}\left(y_i^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_i}{L}\right) \ . \end{aligned} \tag{6.7}$$

The first inequality above is true because $y^{(k,j-1)}$ is a supersolution (by the induction assumption and Lemma 4.3). The second follows from (6.5) by applying a scalar shrinkage on both sides. Combining (6.6) and (6.7), we get

$$y^{(k,j)} \geq \mathbf{S}_{\lambda/L}\left(y^{(k,j)} - \frac{\nabla f(y^{(k,j)})}{L}\right),$$

which proves that $y^{(k,j)}$ is a supersolution. ∎

6.1. Comparison theorem: GD versus CCD. As mentioned below, our strategy for deriving rates for CCD (and CCM) is via a comparison with GD. The result below establishes a comparison theorem for GD versus CCD.

THEOREM 6.2. *Suppose Assumption* 1 *holds and* $\{x^{(k)}\}$ *and* $\{y^{(k)}\}$ *are the sequences of iterates generated by the* GD *and* CCD *algorithms, respectively, when started from the same supersolution* $x^{(0)} = y^{(0)}$. *Then for all* $k \geq 0$

$$y^{(k)} \leq x^{(k)} \ .$$

On the other hand, if they are started from the same subsolution $x^{(0)} = y^{(0)}$, *then the sequences satisfy, for all* $k \geq 0$,

$$y^{(k)} \geq x^{(k)} \ .$$

Proof. We will prove Theorem 6.2 only for the supersolution case by induction on k. The base case is trivial since $y^{(0)} = x^{(0)}$. Now we assume $y^{(k)} \leq x^{(k)}$ and will prove $y^{(k+1)} \leq x^{(k+1)}$. Fix a $j \in [d]$. Note that we have

$$y_j^{(k+1)} = y_j^{(k,j)} = S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) \ .$$

By Proposition 6.1, $y^{(k,j-1)} \leq y^{(k)}$. Applying the isotone operator $\mathbf{S}_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ on both sides and taking the jth coordinate gives

$$S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) \leq S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L}\right) \ .$$

Combining this with the previous equation gives

$$y_j^{(k+1)} \leq S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L}\right) \ . \tag{6.8}$$

Since $y^{(k)} \leq x^{(k)}$ by the induction hypothesis, applying the isotone operator $\mathbf{S}_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ on both sides and taking the jth coordinate gives

$$S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L}\right) \leq S_{\lambda/L}\left(x_j^{(k)} - \frac{[\nabla f(x^{(k)})]_j}{L}\right) \ .$$

By definition,

$$x_j^{(k+1)} = S_{\lambda/L}\left(x_j^{(k)} - \frac{[\nabla f(x^{(k)})]_j}{L}\right) \ . \tag{6.9}$$

Combining this with the previous inequality and (6.8) gives

$$y_j^{(k+1)} \leq x_j^{(k+1)} \ .$$

Since j was arbitrary this means $y^{(k+1)} \leq x^{(k+1)}$, and the proof is complete. ∎

7. Cyclic coordinate minimization. Since CCM minimizes a one-dimensional restriction of the function F, let us define some notation for this subsection. Let

$$f_{|j}(\alpha; x) := f(x_1, \ldots, x_{j-1}, \alpha, x_{j+1}, \ldots, x_d),$$
$$F_{|j}(\alpha; x) := F(x_1, \ldots, x_{j-1}, \alpha, x_{j+1}, \ldots, x_d) \ .$$

With this notation, CCM update can be written as

$$z_j^{(k,j)} = \operatorname*{argmin}_{\alpha} \ F_{|j}(\alpha; z^{(k,j-1)}), \tag{7.1}$$
$$\forall i \neq j, \ z_i^{(k,j)} = z_i^{(k,j-1)} \ .$$

In order to avoid dealing with infinities in our analysis, we want to ensure that the minimum in (7.1) above is attained at a finite real number. This leads to the following assumption.

Assumption 2. For any $x \in \mathbb{R}^d$ and any $j \in [d]$, the one-variable function $f_{|j}(\alpha; x)$ (and hence $F_{|j}(\alpha; x)$) is strictly convex.

This is a pretty mild assumption—considerably weaker than assuming, for instance, that the function f itself is strictly convex. For example, when f is quadratic as in (1.4), then the above assumption is equivalent to saying that the diagonal entries $A_{j,j}$ of the positive semidefinite matrix A are all strictly positive. This is much weaker than saying that f is strictly convex (which would mean A is invertible).

The next lemma shows that the CCM update can be represented in a way that makes it quite similar to the CCD update.

LEMMA 7.1. *Fix $k \geq 0, j \in [d]$ and consider the* CCM *update* (7.1). *Let $g(\alpha) = f_{|j}(\alpha; z^{(k,j-1)})$. If the update is nontrivial, i.e., $z_j^{(k,j)} \neq z_j^{(k,j-1)}$, it can be written as*

$$z_j^{(k,j)} = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{\left[\nabla f(z^{(k,j-1)})\right]_j}{\tau}\right)$$

for

$$\tau = \frac{g'(z_j^{(k,j)}) - g'(z_j^{(k,j-1)})}{z_j^{(k,j)} - z_j^{(k,j-1))}} \ . \tag{7.2}$$

Furthermore, under Assumption 2 *we have $0 < \tau \leq L$.*

Proof. Since $g(\alpha) = f_{|i}(\alpha; z^{(k,j-1)})$ we have

$$g'(\alpha) = \left[\nabla f(z_1^{(k,j-1)}, \ldots, z_{j-1}^{(k,j-1)}, \alpha, z_{j+1}^{(k,j-1)}, \ldots z_d^{(k,j-1)})\right]_j \ .$$

Therefore,

$$g'(z_j^{(k,j-1)}) = [\nabla f(z^{(k,j-1)})]_j \ . \tag{7.3}$$

Since, by definition, $z_j^{(k,j)}$ is the minimizer of $g(\alpha) + \lambda|\alpha|$, we have

$$0 \in g'(z_j^{(k,j)}) + \lambda \operatorname{sign}(z_j^{(k,j)}) \ .$$

For notational convenience we denote $z_j^{(k,j)}$ as $\alpha^\star$, since it is the minimizer of $g(\alpha) + \lambda|\alpha|$. With this notation we have

$$\tau = \frac{g'(\alpha^\star) - g'(z_j^{(k,j-1)})}{\alpha^\star - z_j^{(k,j-1)}} \ . \tag{7.4}$$

Note that τ is well defined since the denominator is nonzero by our assumption of a nontrivial update. Further, $\tau > 0$ by Assumption 2 and $\tau \leq L$ since ∇f (and hence $g'(\alpha)$) is L-Lipschitz continuous.

Depending on the sign of $\alpha^\star$, there are three possible cases.

Case 1: $\alpha^\star > 0$. In this case, we have

$$g'(\alpha^\star) + \lambda = 0 \ . \tag{7.5}$$

By (7.4),

$$g'(\alpha^\star) = g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)}) \ .$$

Plugging this into (7.5), we get

$$g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)}) + \lambda = 0 \ .$$

Using the definition of shrinkage operator (2.5) combined with the fact that $\alpha^\star > 0$, we have

$$\alpha^\star = z_j^{(k,j-1)} - \frac{1}{\tau} g'(z_j^{(k,j-1)}) - \frac{\lambda}{\tau} = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{g'(z_j^{(k)})}{\tau} \right) \ .$$

Case 2: $\alpha^\star = 0$. The corresponding condition is

$$0 \in [g'(\alpha^\star) - \lambda, g'(\alpha^\star) + \lambda] \ .$$

Again using (7.4) and the fact that $\alpha^\star - 0$, we have

$$g'(\alpha^\star) = g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)}) = g'(z_j^{(k,j-1)}) - \tau(z_j^{(k,j-1)}) \ .$$

This can equivalently be written as

$$\alpha^\star = 0 \in \left[\frac{g'(z_j^{(k,j-1)})}{\tau} - z_j^{(k,j-1)} - \frac{\lambda}{\tau}, \frac{g'(z_j^{(k,j-1)})}{\tau} - z_j^{(k,j-1)} + \frac{\lambda}{\tau} \right]$$

or as

$$\alpha^\star = 0 = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{g'(z_j^{(k,j-1)})}{\tau} \right),$$

where the last step follows from the definition of the shrinkage operator (2.5).

Case 3: $\alpha^\star < 0$. In this case, we have

$$g'(\alpha^\star) - \lambda = 0 \ .$$

Using (7.4) to substitute for $g'(\alpha^\star)$ as in the previous cases, we have

$$g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)}) - \lambda = 0,$$

which yields

$$\begin{aligned}\alpha^\star &= z_j^{(k,j-1)} - \frac{1}{\tau} g'(z_j^{(k,j-1)}) + \frac{\lambda}{\tau} \\ &= S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{g'(z_j^{(k,j-1)})}{\tau}\right),\end{aligned}$$

where the last inequality follows because $\alpha^\star < 0$.

Now, using (7.3), we see that

$$z_j^{(k,j)} = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{\left[\nabla f(z^{(k,j-1)})\right]_j}{\tau}\right)$$

holds in all three cases. ∎

We point out that this lemma is useful only for the *analysis* of CCM and not for its implementation (as τ depends recursively on $z_j^{(k,j)}$) except in an important special case. In the quadratic example (1.4), $g(\alpha)$ is a one-dimensional quadratic function. In this case τ does not depend on $z_j^{(k,j)}$ and is simply $A_{j,j}$. This leads to an efficient implementation of CCM for quadratic f.

We are now equipped with everything to prove the following result about the behavior of the CCM iterates.

PROPOSITION 7.2. *Suppose Assumptions* 1 *and* 2 *hold. If z_0 is a supersolution and $\{z^{(k)}\}$ is the sequence of iterates generated by the* CCM *algorithm, then for all $k \geq 0$*

$$(1) \quad z^{(k+1)} \leq z^{(k)}, \qquad (2) \quad z^{(k)} \text{ is a supersolution.}$$

If z_0 is a subsolution and $\{z^{(k)}\}$ is the sequence of iterates generated by the CCD *algorithm, then for all $k \geq 0$*

$$(1) \quad z^{(k+1)} \geq z^{(k)}, \qquad (2) \quad z^{(k)} \text{ is a subsolution.}$$

Proof. See Appendix A. ∎

7.1. Comparison theorem: CCD versus CCM. We already have a comparison theorem comparing GD with CCD (Theorem 6.2). Now, we provide a result for CCD versus CCM.

THEOREM 7.3. *Suppose Assumptions* 1 *and* 2 *hold and $\{y^{(k)}\}$ and $\{z^{(k)}\}$ are the sequences of iterates generated by the* CCD *and* CCM *algorithms, respectively, when started from the same supersolution $y^{(0)} = z^{(0)}$. Then for all $k \geq 0$*

$$z^{(k)} \leq y^{(k)} .$$

On the other hand, if they are started from the same subsolution $y^{(0)} = z^{(0)}$, then the sequences satisfy, for all $k \geq 0$,

$$z^{(k)} \geq y^{(k)} .$$

Proof. See Appendix B. ∎

8. Convergence rates. Our results so far have provided inequalities comparing the *iterates* generated by the three algorithms. We finally want to compare the *objective function values* obtained by these iterates. To do this, the next lemma is useful.

LEMMA 8.1. *If y is a supersolution and $y \leq x$, then $F(y) \leq F(x)$.*

Proof. Since F is convex, we have

$$F(y) - F(x) \leq \langle \nabla f(y) + \lambda\rho, y - x \rangle \tag{8.1}$$

for any $\rho \in \partial\|y\|_1$. We have assumed that $y \leq x$. Thus in order to prove $F(y) - F(x) \leq 0$, it suffices to show that

$$\forall i \in [d], \qquad \exists \rho_i \in \operatorname{sign}(y_i) \qquad \text{s.t.} \qquad \gamma_i + \lambda\rho_i \geq 0 \ , \tag{8.2}$$

where, for convenience, we denote the gradient $\nabla f(y)$ by γ. Since y is a supersolution, Lemma 4.3 gives

$$\forall i \in [d], \qquad y_i \geq S_{\lambda/L}\left(y_i - \frac{\gamma_i}{L}\right) \ . \tag{8.3}$$

For any $i \in [d]$, there are three mutually exclusive and exhaustive cases.

Case 1: $y_i > \frac{\gamma_i + \lambda}{L}$. Plugging this value into (8.3) and using the definition of scalar shrinkage (2.5), we get

$$y_i \geq y_i - \frac{\gamma_i + \lambda}{L} \ ,$$

which gives $\gamma_i + \lambda \geq 0$ and hence $y_i > 0$. Thus, we can choose $\rho_i = 1 \in \operatorname{sign}(y_i)$, and we indeed have $\gamma_i + \lambda\rho_i \geq 0$.

Case 2: $y_i \in [\frac{\gamma_i - \lambda}{L}, \frac{\gamma_i + \lambda}{L}]$. In this case, we have $y_i \geq S_{\lambda/L}(y_i - \frac{\gamma_i}{L}) = 0$. Thus,

$$\frac{\gamma_i + \lambda}{L} \geq y_i \geq 0 \ .$$

Thus we can choose $\rho_i = 1 \in \operatorname{sign}(y_i)$, and we have $\gamma_i + \lambda\rho_i \geq 0$.

Case 3: $y_i < \frac{\gamma_i - \lambda}{L}$. Plugging this value into (8.3) and using the definition of scalar shrinkage (2.5), we get

$$y_i \geq y_i - \frac{\gamma_i - \lambda}{L} \ ,$$

which gives $\gamma_i - \lambda \geq 0$. Now if $y_i \leq 0$, we can set $\rho = -1 \in \operatorname{sign}(y_i)$ and have $\gamma_i + \lambda\rho_i \geq 0$. On the other hand, if $y_i > 0$, we need to choose $\rho_i = 1$, and thus $\gamma_i + \lambda \geq 0$ should hold if (8.2) is to be true. However, we know $\gamma_i - \lambda \geq 0$, and $\lambda \geq 0$, so $\gamma_i + \lambda \geq 0$ is also true.

Thus, in all three cases, there is a $\rho_i \in \operatorname{sign}(y_i)$ such that (8.2) is true. ∎

There is a similar lemma for subsolutions whose proof, being similar to the proof above, has been omitted.

LEMMA 8.2. *If y is a subsolution and $y \geq x$, then $F(y) \leq F(x)$.*

If we start from a supersolution, the iterates for CCD and CCM always maintain the supersolution property. Thus, Lemma 8.1 ensures that starting from the same initial iterate, the function values of the CCD and CCM iterates always remain less than the function values of the corresponding GD iterates. Since the GD algorithm

has $O(1/k)$ accuracy guarantees according to Theorem 3.1, the same rates must hold true for CCD and CCM. This is formalized in the following theorem.

THEOREM 8.3. *Starting from the same super- or subsolution* $x^{(0)} = y^{(0)} = z^{(0)}$, *let* $\{x^{(k)}\}$, $\{y^{(k)}\}$, *and* $\{z^{(k)}\}$ *denote the* GD, CCD, *and* CCM *iterates, respectively. Under Assumptions* 1 *and* 2, *for any minimizer* $x^\star$ *of* (1.2), *and for all* $k \geq 1$,

$$F(z^{(k)}) \leq F(y^{(k)}) \leq F(x^{(k)}) \leq F(x^\star) + \frac{L\|x^\star - x^{(0)}\|^2}{2\,k} \ .$$

Proof. The theorem follows immediately by combining the comparison theorems (Theorems 6.2 and 7.3) and Lemmas 8.1 and 8.2 with the GD guarantee (Theorem 3.1). ∎

9. Numerical experiments. We perform a set of experiments to compare the objective function values produced by gradient descent (GD), cyclic coordinate descent (CCD), and a stochastic variant of coordinate descent (SCD) as described in [21]. We run these algorithms on the lasso problem (1.3) with the purpose of numerically investigating the two intuitive claims that motivated this work: (i) CCD should outperform GD, and (ii) the performance of CCD should be similar to that of SCD.

We generate the data in the same manner as described in section 5 of the work of Friedman et al. [10]. In particular, we generated $n \times d$ matrices X by sampling the n rows from a d-dimensional mean zero multivariate Gaussian distribution. The pairwise correlation between any two of the d dimensions was set to be 0.3. The values for n and d that we chose are given in Table 9.1. The observations were generated using the linear model

$$y = Xx + \alpha \cdot Z \ , \tag{9.1}$$

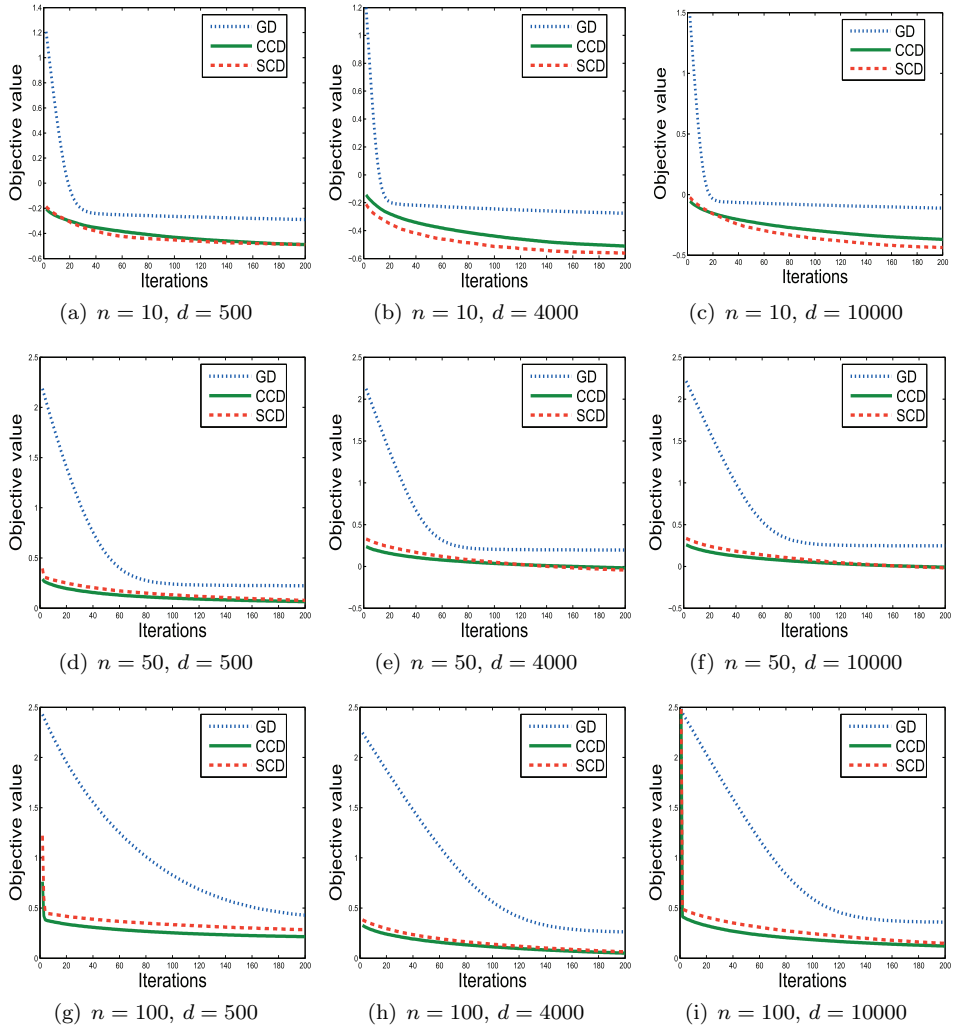
where x has rapidly decaying entries $x_j = (-1)^j \exp(-2(j-1)/20)$ and Z is a multivariate Gaussian noise vector with zero mean and unit variance. The coefficient α is chosen so that the signal-to-noise ratio (SNR) is 3.0.

TABLE 9.1
Sizes of the datasets used in the experiments.

Parameters	Values		
n	10	50	100
d	500	4000	10000

As in Algorithm 1, the step size at every iteration of GD is chosen to be $1/L$, where L is the Lipschitz continuity constant of the smooth part of the objective. For the lasso objective (1.3), L becomes the spectral norm of the covariance matrix $X^\top X$. However, for CCD and SCD, a larger step size still guarantees descent. We follow the choice made by Shalev-Shwartz and Tewari [21] and set the step size for CCD and SCD to be $1/L_1$, where L_1 is the Lipschitz continuity constant of single-dimensional derivatives of the smooth part of the objective (see section 10 for a formal definition). For the lasso objective (1.3), L_1 turns out to be the maximum diagonal element of $X^\top X$.

We set the regularization parameter λ to be 0.1 and run 200 iterations of each of the three algorithms to compare the lasso objectives produced by them. Since one update of either CCD or SCD just updates a single coordinate of x, an "iteration" is defined as d consecutive updates for both CCD and SCD.



(a) $n = 10$, $d = 500$ (b) $n = 10$, $d = 4000$ (c) $n = 10$, $d = 10000$

(d) $n = 50$, $d = 500$ (e) $n = 50$, $d = 4000$ (f) $n = 50$, $d = 10000$

(g) $n = 100$, $d = 500$ (h) $n = 100$, $d = 4000$ (i) $n = 100$, $d = 10000$

FIG. 9.1. *Plots showing logarithm of lasso objective values against number of iterations for three algorithms:* GD*,* CCD*, and* SCD*.* CCD *performs better than* GD *and is comparable to* SCD *across all values of* n *and* d *that we considered.*

The objective values of the three algorithms over 200 iterations are shown in log scale in Figure 9.1. It is interesting to note that we started from the same arbitrary initial point $x = \mathbf{0}$ for all three algorithms. Clearly, this point is not necessarily a supersolution (or subsolution) for the randomly generated data. Despite starting from an arbitrary point, we observe that the objective values produced by CCD are uniformly lower than the objective values obtained by running GD. Moreover, SCD produces objective values comparable to those of CCD, but both of them are significantly better than GD. This reinforces our belief that CCD should enjoy a nonasymptotic convergence rate of $O(1/k)$, similar to those enjoyed by GD and SCD, regardless of the assumptions regarding isotonicity and starting from a super- or subsolution. These assumptions are probably just artifacts of our proof technique, whereas the experiments make us feel confident that nonasymptotic guarantees for CCD should be possible under much more general conditions.

10. Discussion. Recently, coordinate descent–based methods have seen a resurgence of popularity in the statistics, signal processing, and machine learning communities due to the simplicity and speed of the updates when dealing with large scale and high-dimensional data. However, nonasymptotic convergence rates for various types of coordinate descent methods are just beginning to be understood. We now summarize a few recent papers that have shed new light on such convergence rates. This will not only help us situate our work with respect to existing literature, but will also indicate avenues for future work.

In the discussion below, "convergence rate" refers to a nonasymptotic one. Tseng and Yun's work [23] appears to be the first to provide convergence rates for coordinate descent for solving smooth plus separable convex minimization problems. For their convergence rate analysis, they consider minimization of a function $F = f + P$, where f is smooth and convex and P is convex and separable. In fact, they additionally assume that there are m linear equality constraints. So, the setting of this paper would be the special case when $m = 0$ and $P(x) = \lambda\|x\|_1$. Also, they analyze *block* coordinate descent where more than one coordinate can be simultaneously selected and updated. They call their selection rule the "Gauss–Southwell q-rule." For vanilla coordinate descent, i.e., when the block size is one, their accuracy guarantee (obtained by combining Theorem 5.1 and Proposition 6.1 in [23]) after k iterations is

$$O\left(\frac{dL_1 R_{TY}^2}{k}\right), \tag{10.1}$$

where $L_1 = \max_{j\in[d]} L_{1,j}$ is the Lipschitz continuity constant for single-dimensional derivatives of f. That is, whenever x and x' agree in all coordinates except j, we have

$$f(x) \le f(x') + \langle \nabla f(x'), x - x' \rangle + \frac{L_{1,j}}{2}|x_j - x'_j|^2 \ .$$

Note that we always have $L_1 \le L$, but L_1 can be much smaller than L in some cases. The constant R_{TY} measures how far the starting point is from a minimizer $x^\star$:

$$R_{TY} = \max \left\{ \|x - x^\star\| \ : \ F(x) \le F(x^{(0)}) \right\} \ .$$

Shalev-Shwartz and Tewari [21] analyze stochastic coordinate descent (SCD), a randomized version of coordinate descent where the selection of a coordinate is made at random. Shalev-Shwartz and Tewari fix the probabilities to be uniform over coordinates and show an expected accuracy bound of

$$O\left(\frac{d(L_1\|x^\star - x^{(0)}\|^2 + F(x^{(0)}))}{k}\right) \tag{10.2}$$

after k randomized selection steps. We note that only $P = \lambda\|\cdot\|_1$ is considered in [21], but the proof techniques therein easily extend to any convex separable P. Even though (10.1) and (10.2) look similar, it is important to remember that the Gauss–Southwell q-rule typically takes $O(d)$ time to implement. The $O(1)$ implementation of the randomized rule was one of the main motivations for its consideration in [21]. Another motivation behind Shalev-Shwartz and Tewari's choice of uniform probabilities was a justification for the good practical performance of CCD. The intuitive (but, so far, not rigorously justified) reasoning behind this justification is that the randomized rule should behave similarly to the cyclic rule.

Independently of [21], Nesterov [17] also analyzed randomized versions of coordinate descent for solving smooth convex minimization problems. He considered blocks

of arbitrary sizes and nonuniform probabilities for selecting a block at random. There are several results in his paper, but the one most relevant to our discussion here is his Theorem 1. Specializing to block size 1 and uniform probabilities, the rate in that theorem becomes

$$O\left(\frac{dR_N^2}{k}\right), \tag{10.3}$$

where

$$R_N = \max\ \left\{\|x - x^\star\|_{[1]}\ :\ F(x) \le F(x^{(0)})\right\}$$

is defined using the norm

$$\|x\|_{[1]} := \sqrt{\sum_{j\in[d]} L_{1,j}|x_j|^2}\ .$$

Note that we can always upper bound $L_{1,j}$ by L_1, giving us rates comparable to (10.1) and (10.2). But having a bound directly in terms on $L_{1,j}$'s is quite useful if they vary widely.

Richtárik and Takáč [19] provide a nice synthesis by simultaneously generalizing the results in both [21] and [17]. Like [21], their analysis applies to composite nonsmooth functions of the form $f + P$ where the nonsmooth part P is separable. Moreover, they consider probabilities more general than the ones considered in [17].

We wish to alert the reader to the fact that, in the bounds (10.2) and (10.3), the counter k refers to the number of randomized coordinate updates. On the other hand, in Theorem 8.3, k refers to the number of *iterations* of CCD and CCM. Since we have defined an iteration of CCD and CCM to consist of d coordinate updates, the reader can now see why d appears in (10.2) and (10.3) but does not occur in Theorem 8.3.

With the growing interest in large scale data analysis, researchers have naturally gotten interested in parallelization issues. Even though coordinate descent seems inherently sequential, recent work by Bradley et al. [4] has shown that it can be parallelized with linear speedups up to a problem-dependent constant. Their analysis builds on that of Shalev-Shwartz and Tewari [21] and is also nonasymptotic.

Finally, we mention that several "greedy" versions of coordinate descent have appeared in the literature. They all build on the idea of choosing a coordinate greedily to optimize some myopic measure of "progress." Some of these versions, such as the one proposed in [13], have been analyzed from a nonasymptotic viewpoint [7, 20], while others, such as the proposal in [25], still lack nonasymptotic convergence guarantees.

We conclude the paper by reiterating the major open problem concerning cyclic coordinate descent that inspired this work. Through our comparative analysis of GD, CCD, and CCM algorithms, we were able to provide the first known nonasymptotic guarantees for the convergence rates of cyclic coordinate descent methods. However, our results require that the algorithms start from a supersolution (or subsolution) so that the property is maintained for all the subsequent iterates. We also require an isotonicity assumption on the $\mathbf{I} - \nabla f/L$ operator. It is quite desirable to have a more general analysis without any restrictive assumptions. Since stochastic coordinate descent [17, 19, 21] converges at an $O(1/k)$ rate as GD without additional assumptions, intuition and the experiments in section 9 suggest that the same should be true for CCD and CCM. Proving this rigorously remains an open problem.

Appendices. Proofs. The two proofs given below were omitted from the main body of the paper because they were similar to ones already presented.

Appendix A. Proof of Lemma 7.2. Again, we will only prove the supersolution case, as the subsolution case is analogous. We are given that $z^{(0)}$ is a supersolution. We will prove the following: if $z^{(k)}$ is a supersolution, then

$$z^{(k+1)} \leq z^{(k)} \ , \tag{A.1}$$

$$z^{(k+1)} \text{ is a supersolution.} \tag{A.2}$$

Then the lemma follows by induction on k. Let us assume that $z^{(k)}$ is a supersolution and try to prove (A.1) and (A.2). To prove these we will show that $z^{(k,j)} \leq z^{(k)}$ and $z^{(k,j)}$ is a supersolution by induction on $j \in \{0, 1, \ldots, d\}$. This proves (A.1) and (A.2) for $z^{(k+1)}$ since $z^{(k+1)} = z^{(k,d)}$.

The base case ($j = 0$) of the induction is trivial because $z^{(k,0)} \leq z^{(k)}$ since the two vectors are equal. For the same reason, $z^{(k,0)}$ is a supersolution since we have assumed $z^{(k)}$ to be a supersolution. Now assume $z^{(k,j-1)} \leq z^{(k)}$ and $z^{(k,j-1)}$ is a supersolution for some $j > 0$. We want to show that $z^{(k,j)} \leq z^{(k)}$ and $z^{(k,j)}$ is a supersolution. If the update to $z^{(k,j)}$ was trivial, i.e., $z^{(k,j-1)} = z^{(k,j)}$, then there is nothing to prove. Therefore, for the remainder of the proof assume that the update is nontrivial (and hence Lemma 7.1 applies).

Since $z^{(k,j-1)}$ and $z^{(k,j)}$ differ only in the jth coordinate, to show that $z^{(k,j)} \leq z^{(k)}$ given that $z^{(k,j-1)} \leq z^{(k)}$, it suffices to show that $z^{(k,j)} \leq z^{(k,j-1)}$, i.e.,

$$z_j^{(k,j)} \leq z_j^{(k,j-1)} = z_j^{(k)} \ . \tag{A.3}$$

As in Lemma 7.1, let us denote $f_{|j}(\alpha; z^{(k,j-1)})$ by $g(\alpha)$. The lemma gives us a $\tau \in (0, L]$ such that

$$z_j^{(k,j)} = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau}\right) \ . \tag{A.4}$$

Since $z^{(k,j-1)}$ is a supersolution by the induction hypothesis and $\tau \leq L$, using Lemma 4.4 we get

$$\begin{aligned} z_j^{(k,j)} &\leq S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right) \\ &\leq S_{\lambda/L}\left(z_j^{(k)} - \frac{[\nabla f(z^{(k)})]_j}{L}\right) \leq z_j^{(k)}, \end{aligned}$$

where the second inequality above holds because $z^{(k,j-1)} \leq z^{(k)}$ by the induction hypothesis and since $\mathbf{S}_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ is an isotone operator. The third holds since $z^{(k)}$ is a supersolution (coupled with Lemma 4.3). Thus, we have proved (A.3).

We now need to prove that $z^{(k,j)}$ is a supersolution. To this end, we first claim that

$$z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau} = z_j^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_j}{\tau} \ . \tag{A.5}$$

This is true since

$$\begin{aligned} & z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau} - z_j^{(k,j)} + \frac{[\nabla f(z^{(k,j)})]_j}{\tau} \\ &= z_j^{(k,j-1)} - z_j^{(k,j)} - \frac{1}{\tau}(g'(z_j^{(k,j-1)}) - g'(z_j^{(k,j)})) \\ &= z_j^{(k,j-1)} - z_j^{(k,j)} - (z_j^{(k,j-1)} - z_j^{(k,j)}) = 0 \ . \end{aligned}$$

The first equality is true by definition of g and the second by (7.2). Now, applying $S_{\lambda/\tau}$ to both sides of (A.5) and using (A.4), we get

$$\begin{aligned} z_j^{(k,j)} &= S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau}\right) \\ &= S_{\lambda/\tau}\left(z_j^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_j}{\tau}\right) \ . \end{aligned} \tag{A.6}$$

For $i \neq j$, $z_i^{(k,j)} = z_i^{(k,j-1)}$ and thus we have

$$\begin{aligned} & z_i^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_i}{\tau} - z_i^{(k,j)} + \frac{[\nabla f(z^{(k,j)})]_j}{\tau} \\ &= -\frac{1}{\tau}\left[[\nabla f(z^{(k,j-1)})]_i - [\nabla f(z^{(k,j)})]_i\right] \geq 0 \ . \end{aligned}$$

The last inequality holds because we have already shown that $z^{(k,j-1)} \geq z^{(k,j)}$, and thus by isotonicity of $\mathbf{I} - \nabla f/L$ we have

$$[\nabla f(z^{(k,j-1)})]_i - [\nabla f(z^{(k,j)})]_i \leq L(z_i^{(k,j-1)} - z_i^{(k,j)}) = 0 \ .$$

Using the monotonic scalar shrinkage operator, we have

$$S_{\lambda/\tau}\left(z_i^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_i}{\tau}\right) \geq S_{\lambda/\tau}\left(z_i^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_i}{\tau}\right),$$

which, using the inductive hypothesis that $z^{(k,j-1)}$ is a supersolution, further yields

$$\begin{aligned} z_i^{(k,j)} = z_i^{(k,j-1)} &\geq S_{\lambda/\tau}\left(z_i^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_i}{\tau}\right) \\ &\geq S_{\lambda/\tau}\left(z_i^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_i}{\tau}\right) \ . \end{aligned} \tag{A.7}$$

Combining (A.6) and (A.7), we get

$$z^{(k,j)} \geq \mathbf{S}_{\lambda/\tau}\left(z^{(k,j)} - \frac{\nabla f(z^{(k,j)})}{\tau}\right),$$

which proves that $z^{(k,j)}$ is a supersolution.

Appendix B. Proof of Theorem 7.3. We will only prove the supersolution case, as the subsolution case is analogous. Given that $y^{(0)} = z^{(0)}$ is a supersolution, we will prove the following: if $z^{(k)} \leq y^{(k)}$, then

$$z^{(k+1)} \leq y^{(k+1)} \ . \tag{B.1}$$

Then the lemma follows by induction on k. Let us assume $z^{(k)} \leq y^{(k)}$ and try to prove (B.1). To this end we will show that $z^{(k,j)} \leq y^{(k,j)}$ by induction on $j \in \{0, 1, \ldots, d\}$. This implies (B.1) since $z^{(k+1)} = z^{(k,d)}$ and $y^{(k+1)} = y^{(k,d)}$.

The base case ($j = 0$) is true by the given condition in the lemma since $z^{(k,0)} = z^{(k)}$ as well as $y^{(k,0)} = y^{(k)}$. Now, assume $z^{(k,j-1)} \leq y^{(k,j-1)}$ for some $j > 0$. We want to show that $z^{(k,j)} \leq y^{(k,j)}$.

Since $z^{(k,j-1)}, z^{(k,j)}$ and $y^{(k,j-1)}, y^{(k,j)}$ differ only in the jth coordinate, to show that $z^{(k,j)} \leq y^{(k,j)}$ given that $z^{(k,j-1)} \leq y^{(k,j-1)}$, it suffices to show that

$$z_j^{(k,j)} \leq y_j^{(k,j)} \ . \tag{B.2}$$

If the update to $z^{(k,j)}$ is nontrivial, then using Lemma 7.1, there is a $\tau \in (0, L]$, such that

$$\begin{aligned} z_j^{(k,j)} &= S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau}\right) \\ &\leq S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right) \ , \end{aligned} \tag{B.3}$$

where the last inequality holds because of Lemma 4.4 and the fact that $z^{(k,j-1)}$ is a supersolution (Proposition 7.2). If the update is trivial, i.e., $z_j^{(k,j)} = z_j^{(k,j-1)}$, then using (7.1) and (4.2) we have

$$0 \in [\nabla f(z^{(k,j)})]_j + \lambda \operatorname{sign}(z_j^{(k,j)}),$$

which coupled with (4.3) gives

$$z_j^{(k,j)} = S_{\lambda/L}\left(z_j^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_j}{L}\right) \leq S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right),$$

where the last inequality is obtained by applying the isotone operator $\mathbf{S}_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ to the inequality $z^{(k,j)} \leq z^{(k,j-1)}$, which holds by Proposition 7.2. Thus (B.3) holds irrespective of the triviality of the update.

Now applying the same isotone operator to the inequality $z^{(k,j-1)} \leq y^{(k,j-1)}$ and taking the jth coordinate gives

$$S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right) \leq S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) \ .$$

The right-hand side above is, by definition, $y_j^{(k,j)}$. So, combining the above with (B.3) gives (B.2) and proves our inductive claim.

REFERENCES

[1] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[2] M. Belkin, P. Niyogi, and V. Sindhwani, *Manifold regularization: A geometric framework for learning from labeled and unlabeled examples*, J. Mach. Learn. Res., 7 (2006), pp. 2399–2434.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Upper Saddle River, NJ, 1989.

[4] J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin, *Parallel coordinate descent for L_1-regularized loss minimization*, in Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 321–328.
[5] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data*, Springer Ser. Statist., Springer, Heidelberg, 2011.
[6] F. R. K. Chung, *Spectral Graph Theory*, CBMS Regional Conf. Ser. in Math. 92, AMS, Providence, RI, 1997.
[7] I. S. Dhillon, P. Ravikumar, and A. Tewari, *Nearest neighbor based greedy coordinate descent*, in Advances in Neural Information Processing Systems 24, 2011, pp. 2160–2168.
[8] J. Duchi and Y. Singer, *Efficient learning using forward-backward splitting*, in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, eds., 2009, pp. 495–503.
[9] Y. C. Eldar and G. Kutyniok, eds., *Compressed Sensing: Theory and Applications*, Cambridge University Press, Cambridge, UK, 2012.
[10] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, *Pathwise coordinate optimization*, Ann. Appl. Statist., 1 (2007), pp. 302–332.
[11] J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for generalized linear models via coordinate descent*, J. Statist. Software, 33 (2010), pp. 1–22.
[12] A. Genkin, D. D. Lewis, and D. Madigan, *Large-scale Bayesian logistic regression for text categorization*, Technometrics, 49 (2007), pp. 291–304.
[13] Y. Li and S. Osher, *Coordinate descent optimization for ℓ_1 minimization with application to compressed sensing: A greedy algorithm*, Inverse Problems Imaging, 3 (2009), pp. 487–503.
[14] H. Lutkepohl, *Handbook of Matrices*, Wiley, Chichester, UK, 1997.
[15] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$*, Soviet Math. Dokl., 269 (1983), pp. 543–547.
[16] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic, Boston, 2003.
[17] Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362.
[18] W. C. Rheinboldt, *On M-functions and their application to nonlinear Gauss–Seidel iterations and to network flows*, J. Math. Anal. Appl., 32 (1970), pp. 274–307.
[19] P. Richtárik and M. Takáč, *Iteration Complexity of Randomized Block-Coordinate Descent Methods for Minimizing a Composite Function*, preprint, arXiv:1107.2848v1 [math.OC], 2011.
[20] P. Richtárik and M. Takáč, *Efficient serial and parallel coordinate descent methods for huge-scale truss topology design*, in Operations Research Proceedings 2011, Springer, 2012, pp. 27–32.
[21] S. Shalev-Shwartz and A. Tewari, *Stochastic methods for l_1 regularized loss minimization*, J. Mach. Learn. Res., 12 (2011), pp. 1865–1892.
[22] P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, J. Optim. Theory Appl., 109 (2001), pp. 475–494.
[23] P. Tseng and S. Yun, *A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization*, J. Optim. Theory Appl., 140 (2009), pp. 513–535.
[24] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program. Ser. B, 117 (2009), pp. 387–423.
[25] T. T. Wu and K. Lange, *Coordinate descent algorithms for lasso penalized regression*, Ann. Appl. Statist., 2 (2008), pp. 224–244.